# Integration of robotic resources into FORCEnet

Chinh Nguyen,* Daniel Carroll, and Hoa Nguyen
Space and Naval Warfare Systems Center, San Diego, CA 92152-7383

## ABSTRACT

The Networked Intelligence, Surveillance, and Reconnaissance (NISR) project integrates robotic resources into Composeable FORCEnet to control and exploit unmanned systems over extremely long distances. The foundations are built upon FORCEnet–the U.S. Navy's process to define C4ISR for net-centric operations–and the Navy Unmanned Systems Common Control Roadmap to develop technologies and standards for interoperability, data sharing, publish-and-subscribe methodology, and software reuse.

The paper defines the goals and boundaries for NISR with focus on the system architecture, including the design tradeoffs necessary for unmanned systems in a net-centric model. Special attention is given to two specific scenarios demonstrating the integration of unmanned ground and water surface vehicles into the open-architecture web-based command-and-control information-management system of Composeable FORCEnet. Planned spiral development for NISR will improve collaborative control, expand robotic sensor capabilities, address multiple domains including underwater and aerial platforms, and extend distributive communications infrastructure for battlespace optimization for unmanned systems in net-centric operations.

**Keywords:** robotics, Composeable FORCEnet, net-centric, publish/subscribe

## 1. BACKGROUND

The use of unmanned systems in military applications has exponentially increased in recent years. These are generally standalone systems with different mission domains. They are typically developed in a vertical stovepipe fashion, in which the implementation addresses the design challenges from the lowest physical layer to the user interface level (Figure 1). Historically, horizontal integration of two such systems is performed on a case-by-case basis. [1]

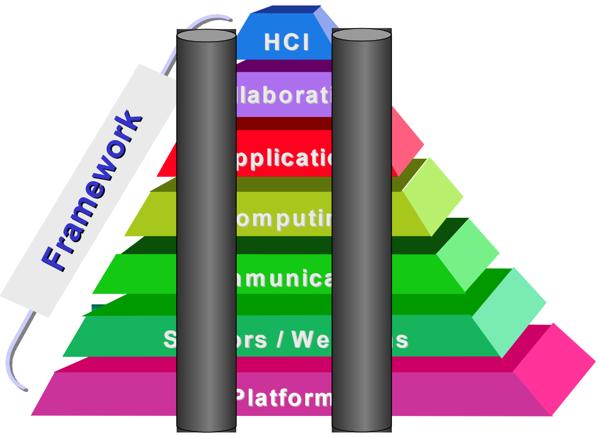


**Figure 1**. The traditional stovepipe model

* Email: *chinh.nguyen@navy.mil*

At the same time, the Navy aims to move towards a net-centric knowledge-based operation known as FORCEnet. Composeable FORCEnet (CFn) is the Space and Naval Warfare System Center San Diego's (SSC San Diego) vision for delivering FORCEnet capability to the Navy. Built on current web technology and industry standards, CFn provides the net-centric infrastructure to support interoperability, collaboration, and data sharing in a publish-and-subscribe methodology. The challenge is to leverage the common framework of CFn to standardize the integration of disparate unmanned systems.

We look at one method to integrate an existing robotic system to CFn. The Networked Intelligence, Surveillance, and Reconnaissance (NISR) project demonstrates the integration of robotic resources to CFn to control and exploit unmanned systems, using unmanned ground vehicles, unmanned surface vehicles, and secure wireless networks. Section 2 will highlight the constraints of the project and the approach taken. Section 3 discusses the implementation, with special focus on the system architecture, including the design tradeoffs necessary for unmanned systems in a net-centric model. Section 3.3 offers two likely scenarios for how unmanned resources will be used in the FORCEnet model.

Finally, Section 4 will discuss planned spiral development for NISR to improve collaborative control, expand robotic sensor capabilities, address multiple domains including underwater and aerial platforms, and extend a distributive communications infrastructure for battlespace optimization for unmanned systems in net-centric operations. The lesson learned from NISR will serve as the foundation for developing a standardize approach to the integration of robotic systems.

## 2. APPROACH

The NISR goal was to integrate unmanned systems into FORCEnet. In particular, the Urban Robot (URBOT) (Figure 2) and the Unmanned Surface Vehicle (USV) (Figure 3) were targeted to expose the challenges of integrating two very different vehicles. Developed by SSC San Diego, both vehicles complied with the Small Robot Technology (SMART) protocol for command and control. [2]

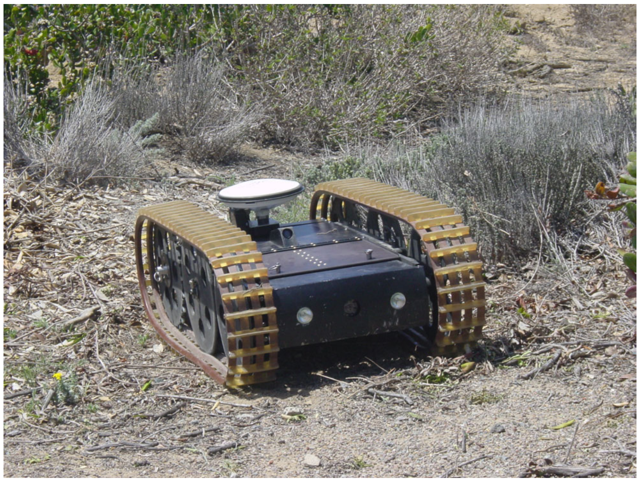


**Figure 2.** SSC San Diego's Urban Robot (URBOT) unmanned ground vehicle (UGV)

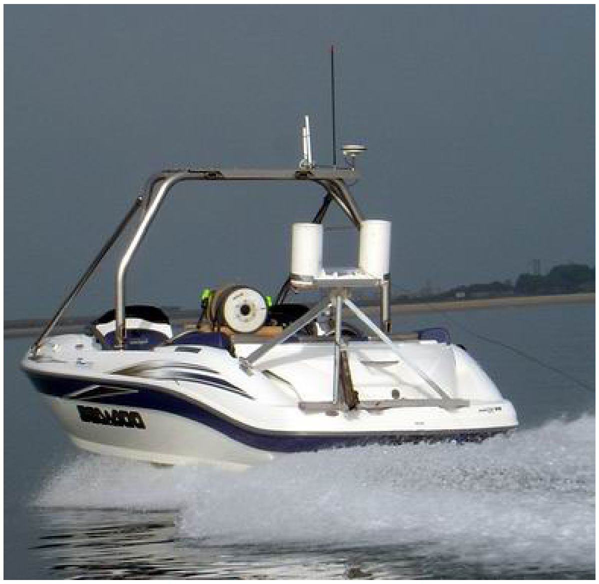


**Figure 3.** SSC San Diego's Unmanned Surface Vehicle (USV)

The physical setup of the effort was an important consideration. In general, we refer to an unmanned system and its associated controller as a local system. The CFn server is located remotely, and the linkage to the local system is open in that no assumption can be made other than the two communicate via TCP/IP. The foremost challenge was to tackle the security concerns and bandwidth requirements for CFn to wirelessly access and control the remote resources. The implementation approach to NISR reflected these issues:

- CFn will not be the main controller of the unmanned resources. The global scale of FORCEnet should not be burdened with the details of command-and-control (C2) of any local system, and FORCEnet acts as a monitory entity. The local C2 system will retain control where the issue of bandwidth requirements is part of the design.
- The interface to CFn will not be real-time. As a monitoring entity, real-time data feed is not required. The data exchange rate between the local system and CFn will be measured in seconds.
- Control of a remote resource will be limited to waypoint navigation. The connectivity between CFn and the local system is assumed to have very limited bandwidth such that teleoperation is not possible.
- Video feed from a remote resource will not be exposed until requested by CFn.
- The wireless network will be secured to the fullest extent.

The next challenge was to determine the data set to publish to CFn, which in the case of the URBOT and the USV, required NISR to identify the common attributes of two very different resources. By extension, these attributes arguably are common to most if not all unmanned vehicles. NISR targeted the following attributes to publish.

- **Robot Status:** All unmanned resources are assumed to have a position and can be moved. The local system will publish robot position, velocity, and heading.
- **Video:** Video feed is standard on almost all unmanned resources. For NISR, the video feed is the ISR data.
- **Local Map Imagery:** Most unmanned systems have more detailed knowledge of the surroundings than the global CFn view. Whether the imagery is preloaded or built on the fly via obstacle mapping, the data will add to the overall battlefield awareness.
- **Robot Control:** The most common control attribute is motion control. Again, the requirement for NISR is waypoint navigation. Although many unmanned vehicles do not yet support waypoint navigation, it is fast becoming a standard feature.

One additional guideline was imposed on NISR: the control interface to the unmanned resources would be web-based in compliance with the net-centric mode of operation that is the cornerstone of FORCEnet.

## 3. ROBOT TO CFn INTEGRATION

Composeable FORCEnet uses the OpenGIS model for geospatial information exchange. The specification for this standard is readily available at http://www.opengeospatial.org. Figure 4 shows the components of CFn in relation to the OpenGIS specification.[3]

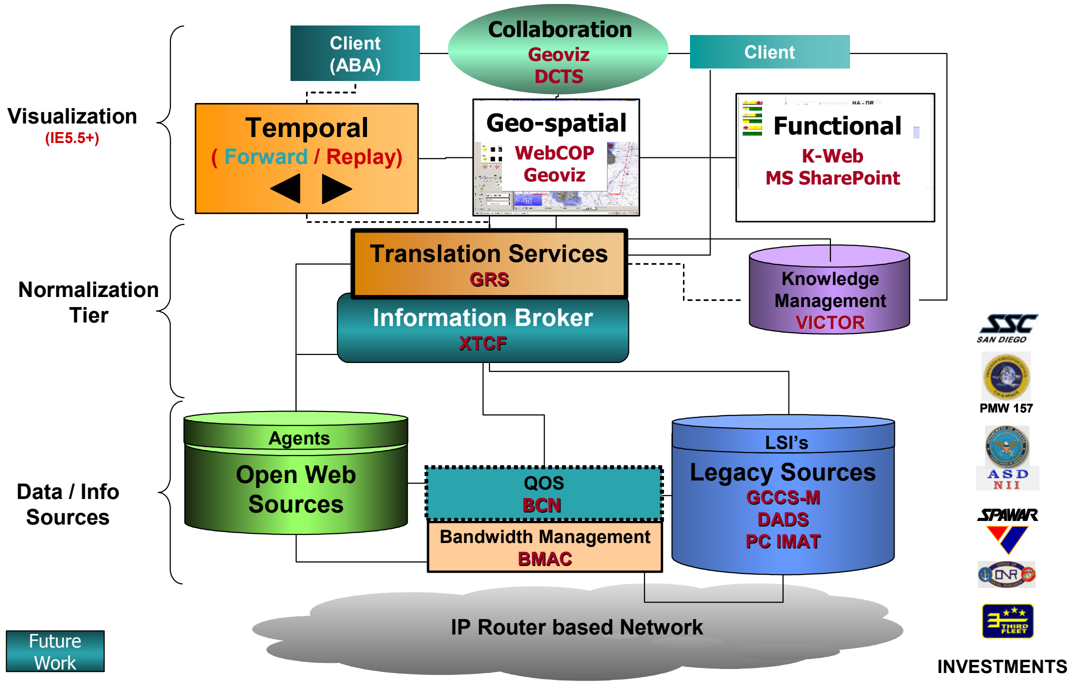


**Figure 4.** Composeable FORCEnet Components

The three tiers reflect the OpenGIS data exchange entities. The data sources publish geospatial data to the normalization tier, which in turn provides services to the visualization tier. The two components relevant to NISR are the Translation Services component, and the Geo-spatial Visualization component. The Translation Services component, labeled *Geospatial Replication Service* (GRS), is the Web Feature Server (WFS) that provides the underlying services to allow for interoperability and data sharing. The Geo-spatial Visualization component, or *Geospatial Collaboration Service* (GCS), subscribes to the GRS component and provides the FORCEnet interface for net-centric operation and collaboration.

The unmanned systems (URBOT and USV) were integrated as data sources in this model, with the implementation performed in two phases. Phase 1 enhanced the existing unmanned systems to add core web capabilities. Starting with the existing SMART architecture, two modules were added to implement a client/server web-based system. Once a working capability was demonstrated, Phase 2 took the final step to publish the robot data to the GRS and to expose the control interface to the GCS. Two scenarios were then developed to exploit the URBOT and USV from the CFn visualization clients.

### 3.1. Phase 1 Development

Phase 1 developed the core web capabilities needed to operate in a net-centric environment. The system design had to answer the challenges brought about by the physical configuration, the monitoring and control of remote wireless robots. Figure 5 shows the design for this phase.

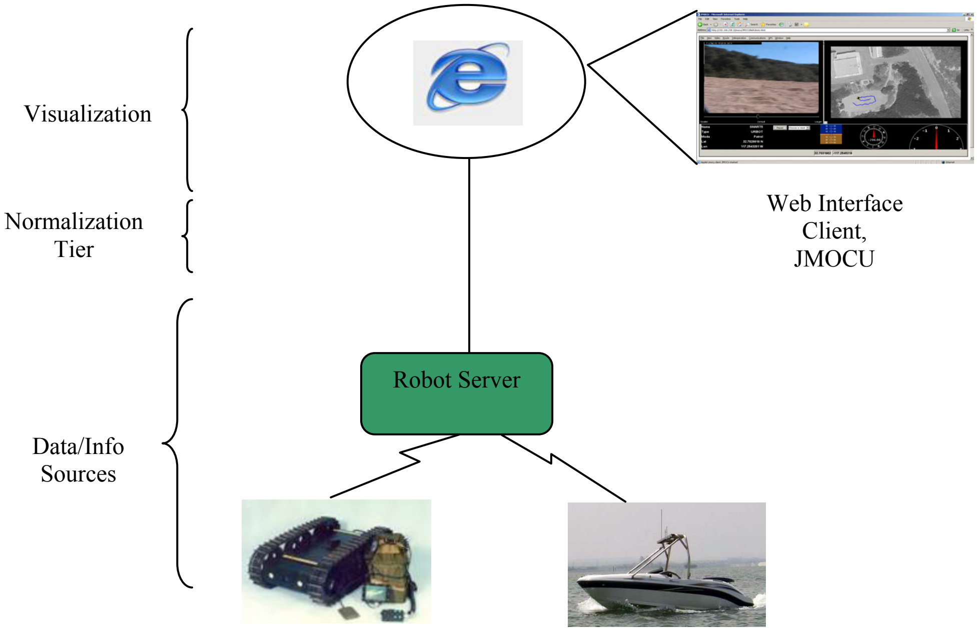
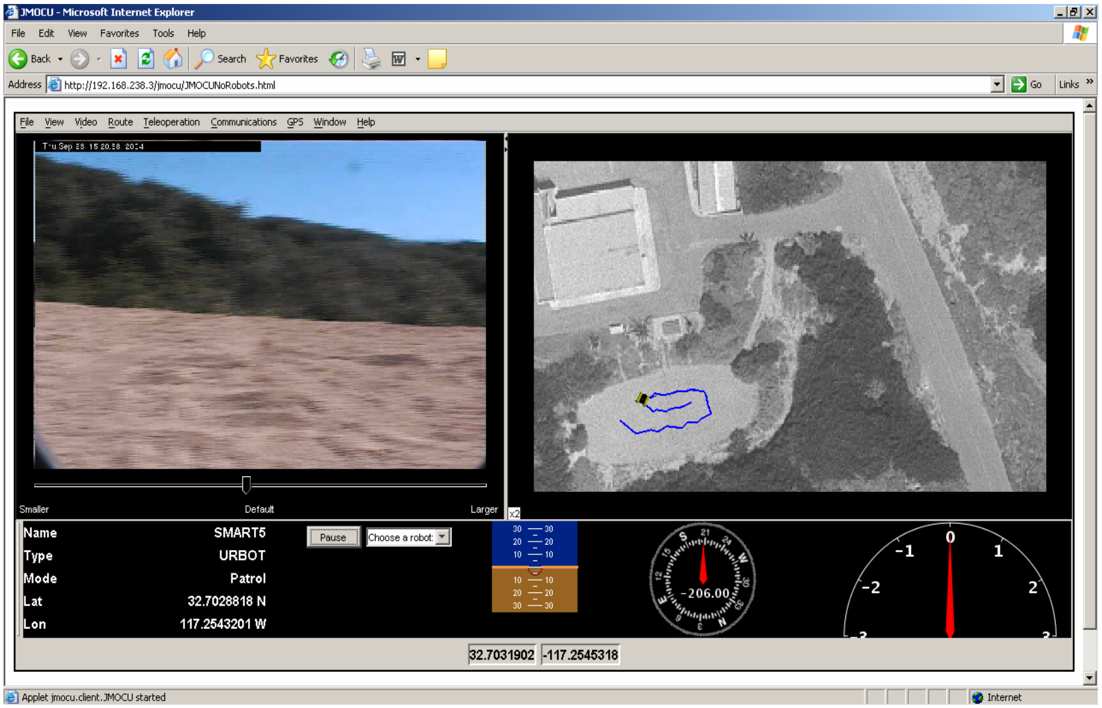


**Figure 5.** NISR Phase 1 Implementation

The Robot Server and the Java-enabled Multi-robot Operator Control Unit (JMOCU) were added to the existing SMART architecture of the local UGV/USV systems in a client/server model. The new components were JAVA implementation over the existing C modules. The programming language selection took advantage of the built-in web technology of JAVA. For example, JMOCU was a JAVA applet using JAVA Remote Method Invocation (JAVA RMI) to access objects within the Robot Server. The client/server components have been successfully tested under both Microsoft Internet Explorer and Mozilla browsers. The Robot Server used JAVA Native Interface (JNI) to the underlying C modules to access the robotic resources. For NISR, JAVA was the right choice with its support for the web and legacy system, in keeping with CFn's theme of using industry standards to maximize interoperability. All for the right price: free!

Ultimately, the Robot Server would be the data source to the GRS component. JAVA also played a key role here during Phase 2, when the JAVA Messaging Service (JMS) was used to publish data to GRS. The choice for this model was selected for several reasons. The Robot Server acted as the single point entry to the underlying unmanned systems, which consist of the URBOT and the USV. Both CFn and the JMOCU go through the Robot Server when requesting robot data and controls. The single-point entry maximized the channel usage to CFn, allowing the addition of more robots without changing the system design. More importantly, it enabled the local system to control access to its resources. For instance, safety logic can be implemented to allow the local user to override a remote user's commands to prevent injury or property damage. Conversely, access logic could be added to allow remote CFn clients to re-task a resource, given the proper authority level. Also, the connection between the Robot Server and CFn is more robust since it need not be wireless. In this fashion, the Robot Server shields CFn from intermittent communication outages frequently experienced by the underlying wireless network. It maintains connectivity and provides continuous data feed from the remaining resources. Even in the worse case of no resource linkage, the Robot Server can provide last known status.

Figure 6 shows the JMOCU running in Microsoft Internet Explorer, using JAVA RMI to access objects in the Robot Server for robot data.


JMOCU - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
Links
File View Video Route Teleoperation Communications GPS Window Help
Thu Sep 28 15 20.58 2004
Smaller
Default
Larger
x2
Pause
Choose a robot:
30 20 10 10 20 30
S 21 24 W 30 33 N 3 6 E 12 15
-206.00
-2 -1 0 1 2 -3 3
Internet

http://192.168.238.3/jmocu/JMOCUNoRobots.html

| | |
|---|---|
| Name | SMART5 |
| Type | URBOT |
| Mode | Patrol |
| Lat | 32.7028818 N |
| Lon | 117.2543201 W |

32.7031902 -117.2545318

Applet jmocu.client.JMOCU started

**Figure 6.** Web Interface Client

JMOCU delivered all four objectives outlined in Section 2. The robot status was provided in the bottom pane, showing the latitude, longitude, and the operation mode of the robot (Patrol Mode shown). The robot heading and speed were shown using the two gauges. The local area map was downloaded from the Robot Server as a JPG image. Waypoint control (the squiggly blue line on the map display) was used to command the remote resource. All requests to execute waypoint paths were made to the Robot Server, which then relayed the commands to the selected robot. The Robot Server provided the video source information, which consisted of the IP address of the onboard camera. In this way, CFn could retrieve the video source information and tap into the video stream without actually taking control of the robot. JMOCU established a direct connection to the source to access the onboard camera.

One of the main concerns with NISR was security of the wireless network. The NISR unmanned resources used standard 802.11 in the 2.4 GHz range. To reduce the security risks, SECNET11 cards were utilized to provide secured linkage to the CFn server (Figure 7).

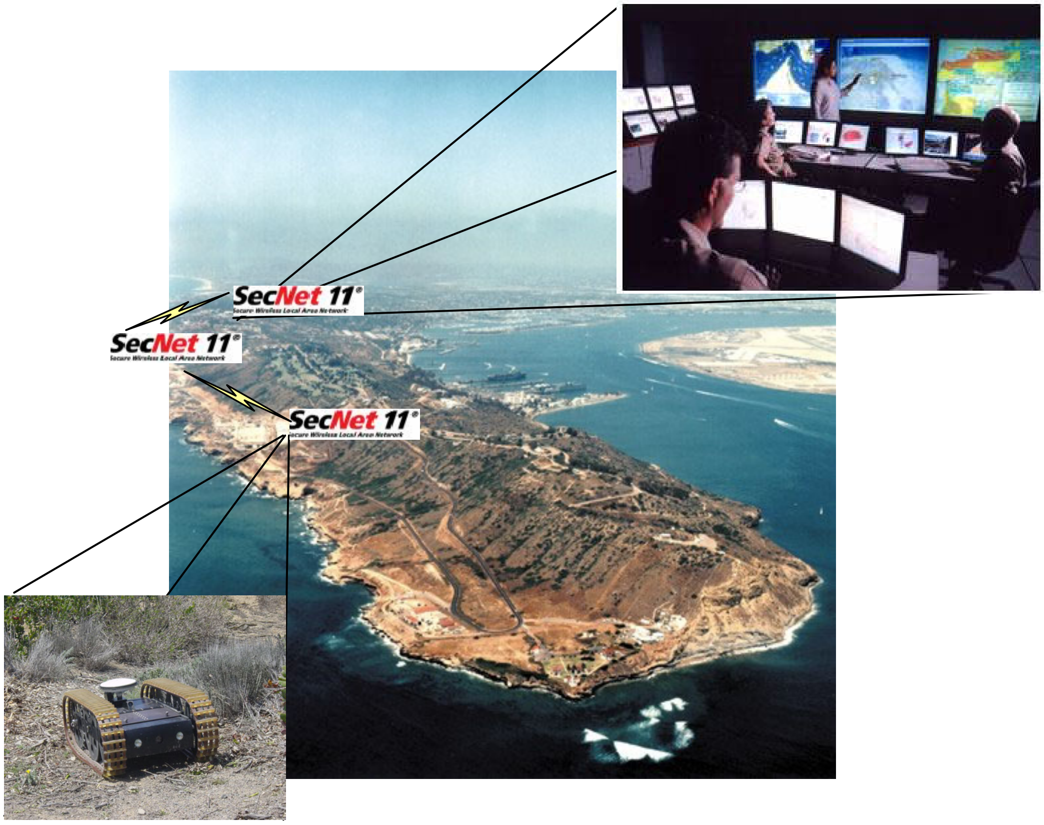

SecNet 11®
Secure Wireless Local Area Network
SecNet 11®
Secure Wireless Local Area Network
SecNet 11®
Secure Wireless Local Area Network

**Figure 7.** Phase 1 Network Configuration

### 3.2. Phase 2 Development

Phase 2 published robotic data from the Robot Server to the CFn's Geospatial Replication Service (GRS). Figure 8 shows the inclusion of the CFn components in the system design.

The link from the Robot Server and the GRS used the JAVA Messaging Service (JMS), an Enterprise Messaging API that came with JAVA and provided a reliable yet loosely coupled communication channel. The Robot Server defined a NISR topic to which the GRS subscribed for data exchange.

The message itself was an Extensible Markup Language (XML) string, standardized by OpenGIS to include Geospatial Markup Language (GML) for robot and system position. However, the Robot Server published additional information to aide CFn's visualization component in its display and operation. For example, the XML data include not only robot status (position, velocity, and heading), but also graphical cues such as icons and textual descriptions. Resources on the unmanned vehicles were also published, as were descriptions of the onboard camera and its IP address were published. Table 1 contains sample data published by the Robot Server. The information is then managed by the GRS and replicated to the visualization entities.

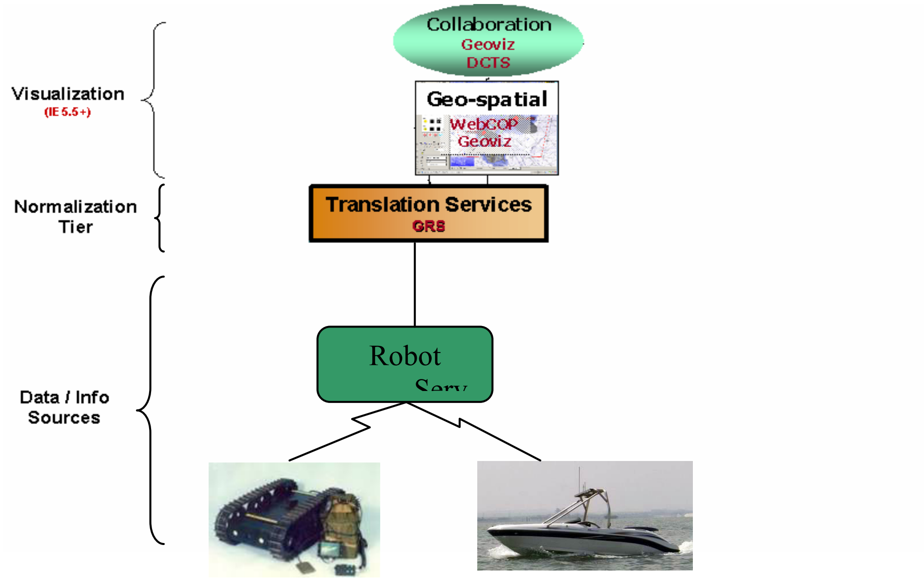


**Figure 8.** NISR Phase 2 Implementation

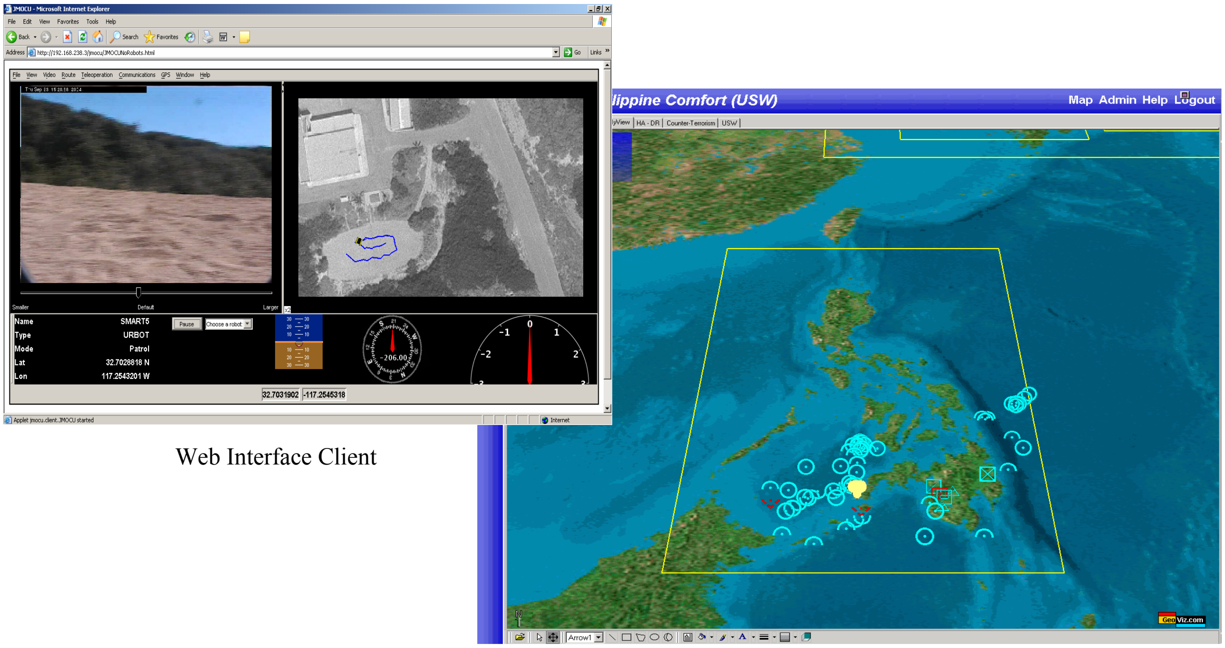
**Table 1.** NISR Robotic Data XML

```
<Resource id="USV9">
   <Name>USV9</Name>
   <Type>USV9</Type>
   <Description>Unmanned Surface Vehicle (simulator)</Description>
   <Mode>Pause</Mode>
   <gml:Point srsName="EPSG:4326">
      <gml:coord>
         <gml:X>120.61401240507068</gml:X>
         <gml:Y>13.860152357980681</gml:Y>
         <gml:Z/>
      </gml:coord>
   </gml:Point>
   <Heading>321.0</Heading>
   <Pitch>8.0</Pitch>
   <Roll>-3.0</Roll>
   <Speed>0.0</Speed>
   <Icon>http://120.0.0.157/jmocu/images/usvLarge.gif</Icon>
   <urlList>
      Unmanned Surface Vehicle (simulator)
      http://120.0.0.157/jmocu/JMOCU.html?resourceName=USV9
      Front Camera
      http://120.0.0.159/view/indexFrame.shtml
   </urlList>
</Resource>
```

## 3.3. NISR SCENARIOS

The design of NISR provided three ways to access the unmanned resources. The most common method was simply to monitor the robotic resources from the CFn visualization tool. The second method was to invoke the Web Interface Client for direct control. Finally, the visualization tool could access just the onboard camera, leaving controls to the local system. With these methods and the features provided by CFn itself, two scenarios were developed to exploit unmanned resources.

In the Direct Control Scenario (Figure 9), the robotic resources are displayed on CFn GeoViz visualization. GeoViz displays the locations and short descriptions of the remote resources. The remote resources are shown as icons over the local area map of the local system. Both icons and the local map are published by the Robot Server. The FORCEnet operator can select a robot, and then invoke the Web Interface Client to take active control. Most likely, this scenario represents the minority of the cases where active control of a particular robot is desired. Figure 10 shows the likely scenario that keyed on the collaborative feature of Composeable FORCEnet.



Name SMART5
Type URBOT
Mode Patrol
Lat 32.7028818 N
Lon 117.2543201 W

32.7031902 -117.2545318

Applet jmocu.client.JMOCU started

Web Interface Client

CFn GeoViz Visualization

**Figure 9.** Direct Control Scenario

In the Collaborative Control Scenario of Figure 10, control is never taken from the local system. The remote FORCEnet operator collaborates with the local FORCEnet entity to request the service of an unmanned system. This is done through CFn's Collaboration view, where data such as maps, geospatial entities, and even screen drawings are shared among the collaboration participants. Figure 10 shows the CFn collaboration view shared by a remote entity in San Diego and a local entity in Japan. It is then up to the local CFn entity to invoke the Web Interface Client to manage and deploy the local unmanned resources to provide the requested service. Alternatively, the local command could engage the local system directly without going through any FORCEnet components. This latter method is more desirable, since the FORCEnet interface is designed to be generic and thus may not offer the full capability available through the native interface. The remote CFn operator can then tap into the onboard camera without controlling the resource directly.

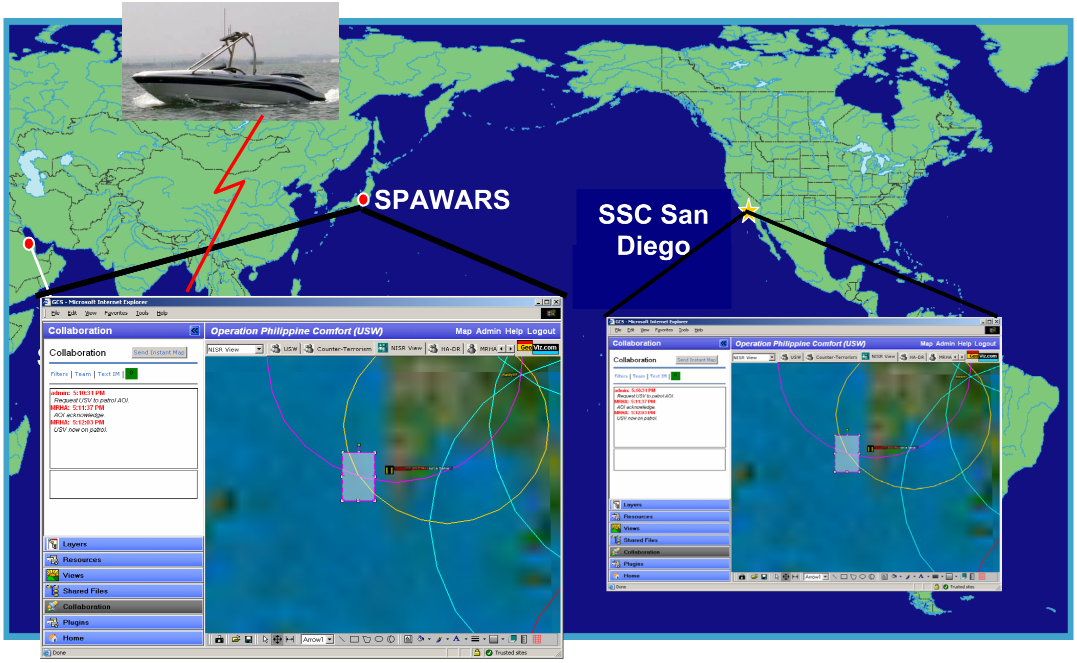
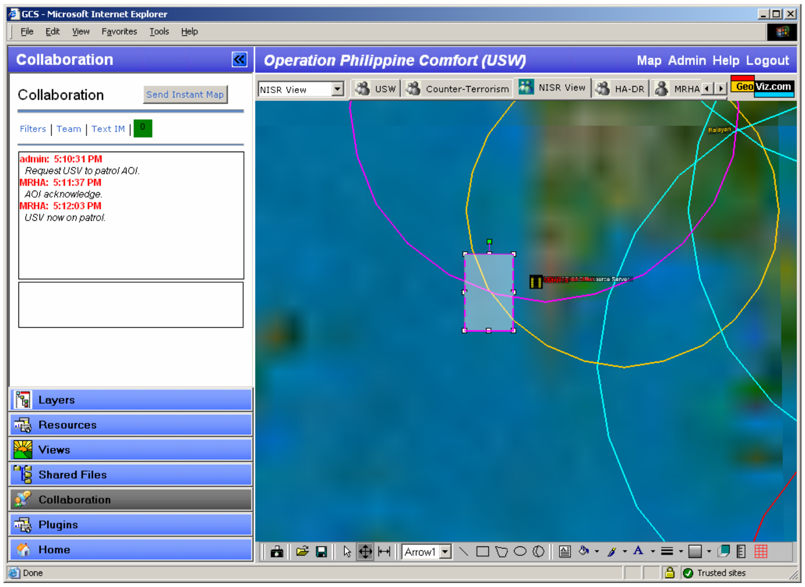


**Figure 10.** Collaborative Control Scenario

Figure 11 shows a detail capture of CFn's collaboration screen. The participants of the session would be looking at the exact same map view. The rectangle area drawn by one operator is shared by all participants. Coordination is also enhanced through text messaging, similar to most Instant Messaging (IM) software.

Both Direct Control and Collaborative Control enable the FORCEnet operator to exploit remote resources, but the Collaborative Control method is more in keeping with the FORCEnet ideal of knowledge-based operation. It is data that is essential in the decision making process. With the data sharing capability available through CFn's Collaborative feature, the Collaborative Control method offers a more attractive option.



admin: 5:10:31 PM
Request USV to patrol AOI.
MRHA: 5:11:37 PM
AOI acknowledge.
MRHA: 5:12:03 PM
USV now on patrol.

**Figure 11.** Composeable FORCEnet Collaboration View

## 4. SPIRAL DEVELOPMENT

Future efforts will build upon the foundation of NISR to provide standardize integration of unmanned systems with FORCEnet. The spiral development will integrate SPAWAR's Multi-robot Operator Control Unit (MOCU) to CFn. MOCU provides supports of multiple messaging protocols, including NATO Standardization Agreement (STANAG) 4586 and the Joint Architecture for Unmanned Systems (JAUS) common standards. Future development will also support more types of unmanned assets, such as unmanned air vehicles (UAVs) and unmanned undersea vehicles (UUVs). These will serve to further enhance situation awareness and mission capability.

The NISR effort highlights the issue of wireless security. Although SECNET11 cards were used to a certain extent, the requirement for these cards to remain in a secure area, or be accompanied by an approved person, makes this method unsuitable for unmanned vehicles. The next iteration needs to include in its design a more secured form of wireless communication. The solution is dependent on developments in this area, but is likely include the use of military radios such the VRC-99.

NISR provides basic waypoint navigation through the Web Interface Client. Future development will likely support a tighter coupling between the FORCEnet visualization and the remote unmanned systems to allow for high level controls. An interface needs to be developed to allow CFn to directly command an unmanned system without invoking a service client. The need to control a specific resource may not be desired. In terms of the publish-subscribe methodology, a service is requested of the unmanned system, which is then responsible for the automated management and deployment of its resources to provide the service. For example, CFn can make a request for the local system to follow the track of a missing ship. The local system is then responsible for determining which of its resources can best perform the task, based upon availability, proximity, or some predefined criteria of the local system. The arrangement frees CFn from the tedium and specific concerns of the resource management, and allows CFn to define and request mission-level tasking.

## 5. CONCLUSION

The NISR effort integrated robotic resources into FORCEnet, relying heavily upon industry standards, including the OpenGIS publish-subscribe standard for geospatial information exchange and the JAVA technology for web development and messaging service. NISR identified the boundary for accessing and controlling remote unmanned resources. It identified the role of FORCEnet as a monitoring entity to the local system, and identified the requirement on the local resources to support waypoint navigation. Two scenarios were offered as ways for FORCEnet to access and exploit unmanned system in a net-centric environment. Most importantly, NISR provided the lessons learned and the foundation upon which spiral development can continue the progression of unmanned systems from standalone stovepipe entities to FORCEnet data sources.

## REFERENCES

1. Smith R. and Grossman J., "Composeable FORCEnet", SPAWAR/NDIA Industry Conference, San Diego, October 2004.

2. Bruch, M.H., Laird, R.T., and Everett H.R., "Challenges for deploying man-portable robots into hostile environments," SPIE Proc. 4195: Mobile Robots XV, Boston, MA, November 5-8, 2000.

3. Nguyen C., Samuel R., Nguyen H.G., and Carroll D., "Unmanned Systems Network-Centric Operations", IDGA Future Naval Plans and Requirements Conference, San Diego, October 2005.